UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/643,577	08/18/2003	Gregory J. Faanes	1376.711US1	3947

21186 7590 07/27/2007

SCHWEGMAN, LUNDBERG, WOESSNER & KLUTH, P.A.
P.O. BOX 2938
MINNEAPOLIS, MN 55402

EXAMINER	
THOMAS, SHANE M	

ART UNIT	PAPER NUMBER
2186	

MAIL DATE	DELIVERY MODE
07/27/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/643,577	FAANES ET AL.	
	Examiner	Art Unit	
	Shane M. Thomas	2186	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *02 May 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *7,8,12,14,15,22,24,25 and 29-32* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *7,8,12,14,15,22,24,25 and 29-32* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1114. Applicant's submission filed on 5/2/2007 has been entered.

Of the previous pending claims, claims 13 and 23 have been canceled, leaving claims 7,8,12,14,15,22,24,25, and 29-32 presently pending.

Response to Arguments/Amendments

Applicant has amended independent claims 7, 22, and 29 to further distinguish the present invention from the prior art reference of Hughes, and in doing so has overcome all pending §102 and §103(a) rejections. A subsequent search of the prior art has uncovered the references of Akkary, Henry, and Merchant, which have been applied with the previous cited Hughes reference to teach the amended claims. New rejections follow.

Claim Objections

Claim 7,8,12,14, and 15 are objected to because of the following informalities:

As per claim 7, the term --the cache-- should be corrected to --the local cache--. Claims 8,12,14, and 15 are objected to as being dependent upon an objected to base claim. Appropriate correction is required.

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claims 24 and 25 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

As per claim 24, it is not clear whether the term --the scalar load/store unit-- refers to --the cache--, the --IRQ--, the --FOQ--, --the scalar processor--, or another element, as the term --the scalar load/store unit-- lack antecedent basis. Nonetheless, for the purposes of examination, the Examiner has considered the term to be --the scalar processor--.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 7,8,12,14,22,24,25, and 29-32 are rejected under 35 U.S.C. 103(a) as being unpatentable over Akkary (U.S. Patent Application Publication No. 2003/0196035) in view of Hughes et al. (U.S. Patent No. 6,393,536) in further view of Henry et al. (U.S. Patent Application Publication No. 2003/0018875) in further view of Merchant et al. (U.S. Patent No. 6,385,715) in further view of Hennessy et al. (*Computer Organization and Design: The Hardware / Software Interface*).

As per claim 7, Akkary teaches:

obtaining a memory request (e.g. store [¶¶30-31] and load [¶32] requests) however Akkary does not specifically teach **storing the memory request in an Initial Request Queue (IRQ)**. Hughes teaches an IRQ (LS1 cache buffer 60) for storing all cache memory requests. Such a system keeps memory latency low for loads that probe the cache [14/32-34]. It would have been obvious to one having ordinary skill in the art at the time the invention was made to have combined the request queue of Hughes with the memory system of Akkary in order to have buffered loads and thereby decrease the memory latency of the system of Akkary.

Akkary further teaches:

processing the memory request from the IRQ by a cache controller (logic inherently associated with the out of execution logic of figure 3 as it is responsible for ultimately accessing the data cache 270 - figure 3) **wherein the processing includes:**

identifying a type of memory request (load or store requests - ¶29);

determining whether the memory request hits in a local cache (¶32 - load requests checked against the data cache 270);

determining whether an address associated with the memory request matches an address in a Forced Order Queue (FOQ) (store buffer 260) - ¶32. Modified Akkary does not specifically teach using only a **portion of the address** to associate with a **partial address** in the FOQ. Henry teaches a store forwarding method (figure 3) that initially compares the index portion of a load address (Step 306) against a pending store's index. Figure 2 shows that the index 204 is a partial address of physical address 188. By comparing only page index in parallel with the TLB lookup, instead of waiting until after the TLB determination to see if the load request address matches a store address in the store buffer ¶31, a reduction in the instruction latency is achieved by reduction in the number of pipeline stages required (¶16). Therefore it would have been obvious to one having ordinary skill in the art at the time the invention was made to have further modified the invention of Akkary with the teachings of Henry in order to have decreased the memory request latencies of accessing the store buffer of Akkary.

Akkary further teaches **if a portion of an address associated with the memory request matches one or more partial addresses in the FOQ** (hitting a store address in the store buffer - ¶32) **preventing the memory request from being satisfied in the local cache** (selector 280 chooses the entry in the store buffer 260 if the load address hits, else the data cache fulfills the

request - ¶32. If the request is fulfilled by the store buffer 260, then the request would be prevented from being fulfilled by the local cache 270 since the selector would not select the request from the data cache 270.)

Akkary does not specifically teach the procedure for fulfilling a memory request when the **memory request misses in the local cache** 270 **and, at the same time,** the **partial addresses in the FOQ** 260. Merchant teaches in [8/13-22] that a load replay queue may be used when a load request misses in the cache [and thereby the store buffer]. By removing a load request that misses the cache and FOQ from the instruction pipeline, the teachings of Merchant do not unnecessarily delay execution of other non-dependent instructions, thereby increasing system throughput and lowering average instruction latency. Further advantages of using a load replay queue are listed in [8/25-39]. Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to have further modified the memory system of Akkary with the teachings of Merchant in order to have removed a load request, and all dependent instructions, from the instruction execution pipeline, thereby increasing instruction throughput. Thus the Examiner is considering the combination of the store buffer 260 of Akkary and the load replay queue of Merchant to be the Forced Order Queue, and as such it would have further been obvious to one of ordinary skill that a queue could store both store requests and pending load requests.

Modified Akkary does not specifically teach **allocating a cache line in the local cache corresponding to the local cache miss**; however, such a step is very well known in the art of caching. Hennessy teaches on page 606 ("Queston 3") that a decision regarding which cache line to remove from the cache, and thereby allocate for the new cache line that is requested,

occurs for a cache miss. Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to have combined the memory system of Akkary with the well known allocation technique of Hennessy in order to have added the requested cache line to the cache by allocating an entry in the cache, thereby increasing the temporal locality of the request. Therefore, the data is stored in the data cache 270 when the data is subsequently requested.

As per claim 8, Akkary teaches **obtaining a memory load** (¶32) **or a memory store request** (¶¶30-31).

As per claim 12, Akkary teachs **processing the memory request using the FOQ when the memory request matches a corresponding request in the FOQ** (store buffer 260) - ¶32. If a match occurs in the FOQ, the selector 280 will select that match instead of the entry from the data cache 270 - ¶32.

As per claim 14, Akkary teaches **processing the memory request in the FOQ** 260 **when local cache processing is bypassed** - ¶32. Cache processing is bypassed as the selector 280 selects the output of the store buffer 260 when a match occurs in the store buffer - ¶32.

As per claims 22 and 29, Akkary teaches a **cache** 270 and a **cache controller** (out of order execution logic which ultimately sends requests to be fulfilled by the cache) **having a FOQ** 260 (figure 3) but does not specifically teach **an IRQ**. Hughes teaches an IRQ (LS1 cache buffer 60) for storing all cache memory requests. Such a system keeps memory latency low for loads that probe the cache [14/32-34]. It would have been obvious to one having ordinary skill in the art at the time the invention was made to have combined the request queue of Hughes with the memory system of Akkary in order to have buffered loads and thereby decrease the memory

latency of the system of Akkary. Thus the IRQ of modified Akkary **buffers a scalar load/store command having a scalar load/store instruction and one or more addresses**. Hughes further teaches that the IRQ **sends the scalar load/store command to the cache controller** [to be executed] [4/23-24] **and the cache** 28.

Akkary teaches **wherein the cache 270 services the load/store command from the IRQ** [of Hughes] **when the scalar load/store command hits in the cache and one of the one or more addresses in the scalar load/store command does not match of or more addresses in the FOQ** (¶32). If the load command misses the FOQ (store buffer 260), the selector 280 selects the requested data as output from the data cache 270 (e.g. a hit from the cache and miss from the FOQ).

Modified Akkary does not specifically teach using only a **portion of the address** to associate with a **partial address** in the FOQ. Henry teaches a store forwarding method (figure 3) that initially compares the index portion of a load address (Step 306) against a pending store's index. Figure 2 shows that the index 204 is a partial address of physical address 188. By comparing only page index in parallel with the TLB lookup, instead of waiting until after the TLB determination to see if the load request address matches a store address in the store buffer ¶31, a reduction in the instruction latency is achieved by reduction in the number of pipeline stages required (¶16). Therefore it would have been obvious to one having ordinary skill in the art at the time the invention was made to have further modified the invention of Akkary with the teachings of Henry in order to have decreased the memory request latencies of accessing the store buffer of Akkary.

Modified Akkary does not specifically teach **adding the scalar load/store command to the FOQ when the scalar load/store command misses in the cache**. As it is well known that caches cannot at one time comprises all the data of a system, cache misses may occur to fetch data from the next lower level of the system's memory. . Merchant teaches in [8/13-22] that a load replay queue may be used when a load request misses in the cache [and thereby the store buffer]. By removing a load request that misses the cache and FOQ from the instruction pipeline, the teachings of Merchant do not unnecessarily delay execution of other non-dependent instructions, thereby increasing system throughput and lowering average instruction latency. Further advantages of using a load replay queue are listed in [8/25-39]. Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to have further modified the memory system of Akkary with the teachings of Merchant in order to have removed a load request, and all dependent instructions, from the instruction execution pipeline, thereby increasing instruction throughput. Thus the Examiner is considering the combination of the store buffer 260 of Akkary and the load replay queue of Merchant to be the Forced Order Queue, and as such it would have further been obvious to one of ordinary skill that a queue could store both store requests and pending load requests.

Modified Akkary does not specifically teach **wherein one or more lines in the cache are allocated for cache line replacement when the scalar load/store command is added to the FOQ and the address for the cache line does not match a partial address in the FOQ**. As in the case above, when the request cannot be fulfilled by the cache 270 of modified Akkary, a cache line fill must occur with the cache line fetched from memory. In order to allocate the new cache line, a previous cache line must be removed. Such a step is very well known in the art

of caching. Hennessy teaches on page 606 ("Queston 3") that a decision regarding which cache line to remove from the cache, and thereby allocate for the new cache line that is requested, occurs for a cache miss. Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to have combined the memory system of Akkary with the well known allocation technique of Hennessy in order to have added the requested cache line to the cache by allocating an entry in the cache, thereby increasing the temporal locality of the request. Therefore, the data is stored in the data cache 270 when the data is subsequently requested.

Further regarding claim 29, Akkary does not specifically teach **a plurality of cache controllers, wherein each cache controller includes a FOQ**. It would have been obvious to one having ordinary skill in the art at the time of invention to have modified the system of Akkary to include a plurality of cache controllers, each with a FOQ, as it has been held that mere duplication of parts has no patentable significance unless a new and unexpected result is produced (MPEP § 2144.04(vi)). As the expected result of claimed invention would not produce a new result whether a single or multiple cache controllers with FOQ were used, such a limitaion would have therefore been obvious.

As per claim 24, Akkary teaches the **scalar load/store unit** (e.g. the processor of figure 3) **includes an address generator** 250 **to generate one or more physical addresses from the one or more addresses of the scalar load/store command** (¶30 and ¶32).

As per claim 25, Henry teaches a **TLB** 104 that is used for translating virtual addresses to physical addresses, as well known in the art, as part of a load to store forwarding mechanism. Refer to figure 1.

As per claim 30, the Examiner is considering the storage element that contains all the entries of the FOQ (e.g. combination of the store buffer 260 of Akkary with the load replay queue of Merchant) to be **a FOQ index array** since the FOQ contains multiple address entries for both the loads (refer to figure 1 of Merchant, element 170) and stores (figure 1, element 110 of Akkary). Further it could be seen by one of ordinary skill that the array could be comprised of only address indexes as the teachings of Henry use only indexes to determine a hit in the load/store buffer (figure 3, steps 306,308).

As per claims 31 and 32, **the FOQ is divided logically into a first queue and second queue where the first queue manages requests to memory** (e.g. the addresses of pending loads and stores as taught in ¶30 of Akkary and [8/13-22] of Merchant). The Examiner is considering the second queue to be the logical division of the entries of the store buffer portion 260 of the FOQ which contain the **data that is to be stored to cache** 270 - ¶30 of Akkary.

Claim 15 is rejected under 35 U.S.C. 103(a) as being unpatentable over Akkary (U.S. Patent Application Publication No. 2003/0196035) in view of Hughes et al. (U.S. Patent No. 6,393,536) in further view of Henry et al. (U.S. Patent Application Publication No. 2003/0018875) in further view of Merchant et al. (U.S. Patent No. 6,385,715) in further view of Hennessy et al. (*Computer Organization and Design: The Hardware / Software Interface*), as applied to claim 7,8,12,14,22,24,25, and 29-32, above, in further view of Yamahata (U.S. Patent No. 5,247,639).

As per claim 15, Akkary does not specifically teach **processing the memory request in the FOQ when the memory request includes a synchronization request that causes local**

cache processing to be bypassed. Hughes teaches a multiprocessing system in figure 13 and [32/47-56] with processors 10 and 10a independently connected to the bus bridge 202 for connection to main memory 204. It would have been obvious to one having ordinary skill in the art to have seen that such a system could be used to practice the store-forwarding techniques of modified Akkary (¶32). Yamahata teaches a cache bypass bit for use when multiple processors are to obtain synchronization by using semaphore data in [2/4-38]. Specifically Yamahata teaches in [2/15-19] that an instruction decoder sends a bypass request to a bus control unit to bypass a local cache. Modified Akkary, utilizing the system of Hughes, shows a bus interface unit 37 connected to the load/store unit 26 in figure 2 of Hughes. Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to have further modified the system of Akkary with the cache bypassing during multiprocessor synchronization teaching of Yamahata in order to have been able to maintain a level of cache coherency between the processors 10 and 10a of modified Akkary when both processors would be attempting to update main memory 204. The utilization of semaphore data instructions are well known in the art to be utilized in multiprocessing systems for contention of a shared resource (in the case of Hughes, it would be main memory 204).

Conclusion

Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Shane M. Thomas whose telephone number is (571) 272-4188. The Examiner can normally be reached M-F 8:30 - 5:30.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Matt M. Kim can be reached at (571) 272-4182. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Shane M. Thomas

MATTHEW KIM
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100